1. A computer system comprising:

a memory;

a register file coupled to the memory through a memory channel, the register file to store data for one or more procedures in one or more frames, respectively; and

a register stack engine to monitor activity on the memory channel and to transfer data between selected frames of the register file and the memory responsive to available bandwidth on the memory channel.

2. The computer system of claim 1, wherein the memory includes a backing store and the register stack engine transfers data between the selected frames and the backing store.

3. The computer system of claim 1, wherein a portion of the register file is organized as a register stack.

4. The computer system of claim 3, wherein the register stack engine includes a first pointer to indicate a first location in a current frame of the register stack.

5. The computer system of claim 4, wherein the register stack engine includes a second pointer to indicate an oldest dirty register in the register stack.

6. The computer system of claim 5, wherein the register stack engine includes a third pointer to indicate an oldest clean register in the register stack.

7. The computer system of claim 1, wherein registers of the register file are mapped to a current frame and an inactive frame, and the register stack engine transfers data between registers in the inactive frame and the backing store.

8. The computer system of claim 7, wherein the registers mapped to the inactive frame are designated as clean or dirty, according to whether data stored in the registers has or has not been spilled to the memory.

9. The computer system of claim 8, wherein the memory includes a backing store.

10. The computer system of claim 9, wherein the register stack engine transfers data from a dirty registers to a corresponding location in the backing store when bandwidth is available on the memory channel.

11. The computer system of claim 9, wherein the register stack engine transfers data to a clean register from a corresponding location in the backing store when bandwidth is available on the memory channel.

12. A method for managing data in a register stack comprising:

designating registers in the register stack as clean or dirty, according to whether data in the registers has been spilled to a backing store;

monitoring operations on a memory channel; and

spilling data from a current oldest dirty register to the backing store when capacity is available on the memory channel.

13. The method of claim 12, further comprising updating a first pointer to indicate a new oldest dirty register when data is spilled from the curent oldest dirty register.

14. The method of claim 12, further comprising filling data from the backing store to a current oldest clean register when capacity is available on the memory channel.

15. The method of claim 14, further comprising updating a second pointer to indicate a new oldest clean register when data is filled to the curent oldest clean register.

16. A computer system comprising:

a memory system;

a register file to store data for an active procedure and one or more inactive procedures; and

a register stack engine to transfer data between registers associated with the one or more inactive procedures and the memory system, responsive to available bandwidth to the memory system.

17. The computer system of claim 16, wherein the computer system further comprises a load/store unit and the register stack engine monitors the load/store unit to determine available bandwidth to the memory system.

18. The computer system of claim 16, wherein the register stack engine includes a first pointer to track a next inactive register to spill to the memory system and a second pointer to track a next inactive register to fill from the memory system responsive to available bandwidth.

19. The computer system of claim 16, wherein the register stack engine transfers data for inactive procedures responsive to a mode status indicator.

20. The computer system of claim 19, wherein the register stack engine operates in a lazy mode, a store intensive mode, a load intensive mode, or an eager mode according to the mode status indicator.

21. The computer system of claim 19, wherein the mode status indicator is set under software control responsive to a type of application to run on the computer system.